

REMARKS

Claims 1-28 are now pending in the application. The Examiner is respectfully requested to reconsider and withdraw the rejection(s) in view of the amendments and remarks contained herein.

REJECTION UNDER 35 U.S.C. § 103

Claims 1-5, 7-14 and 20-28 stand rejected under 35 U.S.C. §103 as being unpatentable over applicant's admitted prior art in view of U.S. Patent No. 5,604,843 (Shaw). This rejection is respectfully traversed.

Shaw relates generally to a method for interfacing with output devices, such as printers. The Examiner believes the universal driver in Shaw is analogous to the multi-role driver and the minidrivers in Shaw are analogous to helper drivers. However, these analogies are inaccurate for various reasons set forth below.

The universal driver plays substantially the *same* role with respect to a number of instances of the same class of device, printers. The multi-role driver plays a substantially different role with respect to several distinct classes of device: SCSI busses, disk drives, and tape drives. The multi-role driver also plays multiple, substantially different roles with respect to one class of device: both lower filter and upper filter for disk drives. Filtering below the disk FDO has an entirely different purpose than filtering above the FDO. Therefore, the Shaw universal driver does not have multiple roles in the sense that the multi-role driver does. In this regard, the Examiner's attention is drawn to claims 3, 12, 17, 22 and 26.

The universal driver exposes one set of functions, each function analogous to one listed in the graphics device interface functions list appearing in Shaw Col 2 lines 5-53. Dissimilarly, the multi-role driver exposes several sets of WDM driver functions, one set per role. Of each set, the one function primarily discussed is the “DOPush” function, similar to the AddDevice routine of a traditional WDM driver, the difference being that there can be at most one true AddDevice routine per driver, while there are in fact several DOPush functions in the multi-role driver, one per role. However, in addition to each DOPush function, there may be for each role a distinct function for handling any one of the various IO requests that may be passed to a WDM driver. For example, there may be for each role a distinct function for handling IRP_MJ_PNP requests; similarly there may be for each role a distinct function for handling IRP_MJ_WRITE requests. There are *not* similarly multiple GDI “Enable” functions or “DevBitBlt” functions in the Shaw universal driver.

In contrast, multiple functions, one per role, for handling each type of IO request, are provided for by the standard Windows Driver Model (WDM). In this model, when the OS passes a PDO to a driver’s AddDevice (here DOPush) routine, the driver may decide to participate in handling IO requests for the device, or not. If it so decides, it proceeds to participate by allocating and initializing its own Device Object (DO) and attaching it to the top of (also known as pushing it onto) the stack of DOs rooted in the PDO. Each DO contains a table of function pointers, one for each IO request type. Subsequent IO requests to the device are handled by passing them, according to their type, to one of the functions from the table of each DO in the stack, sequentially, from top to bottom. Because the multi-role driver has multiple distinct DOPush functions, one

per role, it can easily initialize the table of function pointers in DOs established in each with a distinct set of request handler functions. For at least these reasons, the universal driver in Shaw is not analogous to the multi-role driver in the WINDOWS Driver Model environment.

In the Windows Driver Model, a driver's initialization routine is called before any PDOs can be passed to its AddDevice routine. The AddDevice routine *is not* the driver initialization routine. One purpose of the actual initialization routine is to populate fields in a data structure allocated by the OS called the Driver Object (not to be confused with Device Objects; there is one Driver Object loaded driver and exists independently of whether there are any devices with which the driver is associated). One such field in the Driver Object is the AddDevice function pointer. The OS cannot identify or call a driver's AddDevice routine until the driver has provided a pointer to the routine in this field. The innovation of the present application is that a helper driver may provide not a pointer to any of its own functions, but instead to a role-specific DOPush function in the multi-role driver. When the time comes for the OS to pass a PDO to a registered driver's AddDevice routine, it ends up invoking the multi-role driver's DOPush routine directly, and not involving the helper driver at all. Thus, the helper driver receives from the OS only a single invocation of its own initialization routine, and never receives any call that it must forward to the multi-role driver. In Shaw, the minidriver always receives the calls from the OS (GDI) and forwards to the universal driver. In fact, this innovation of the present invention cannot be applied the mini/universal driver model for printer drivers. For this reason, the teachings of Shaw cannot be combined with the Window Driver Model as described in the present invention.

As a basis for combining Shaw with the AAPA, the Examiner further asserts that one would have been motivated by the fact that devices often share similar functions that can be implemented once and used by multiple devices. The minidrivers taught by Shaw would allow for a reduction in complexity of implementation by allowing developers to implement functions once in the universal driver and used by all of the minidrivers associated with the individual devices. To the contrary, the multi-role driver of the present application eliminates a single AddDevice routine shared by several device types and roles as described in our prior art disclosures in favor of several distinct DOPush functions. The multi-role driver does not have a parallel to functions implemented once in the universal driver and used by all minidrivers. Besides the DOPush functions, the multi-role driver's IO request handling routines are distinct and differ substantially in implementation from one role to another. Applicant's claimed invention is clearly not a case of functions implemented once and used by (or for) multiple similar devices as in Shaw.

Applicant's invention achieves a reduction in complexity in a different manner entirely: eliminating role-discovery code from the multi-role driver's AddDevice routine. This code isn't merely moved to the several DOPush functions; it is eliminated entirely. Any particular DOPush function implies, without any question or conditions, a specific role of the multi-role driver, so when a PDO is passed to a DOPush function, the role need not be discovered.

In view of these arguments, Applicant respectfully submits that the Examiner has failed to establish a prima facie case of obviousness as required by *Graham v. John Deere Co.*, 148 USPQ 459 (1966). Specifically, the references must be considered as a

whole and must suggest the desirability and thus the obviousness of making the combination. In this case, the teachings of the references are not analogous for the reasons set forth above. In addition, the motivation for combining these references is not applicable to the present invention. Therefore, Applicant requests the Examiner to reconsider and withdraw the rejections based on this combination of references.

The Examiner's attention is also drawn to claims 3, 12, 17, 22 and 26 which further defines a role. For example, claim 3 recites "wherein a role is determined according to a device type for which the multi-role driver is invoked". Shaw only deals with a single device type: printers. Each minidriver does not map uniquely to a different role of the multi-role driver. Thus, these claims help to further distinguish the helper drivers from the minidrivers disclosed in Shaw. Moreover, the unidriver does not support different functions for different device types as recited in these claims. Since the Examiner asserts that applicant's amendments necessitated the new ground of rejection, then the Examiner must be giving patentable weight to the uni-role aspect of the helper drivers. In this case, applicants respectfully request the Examiner to reconsider and withdraw these rejections. If the Examiner is not giving patentable weight to this aspect of the helper drivers, then applicant asserts the finality of the rejection is improper and should be withdrawn.


CONCLUSION

It is believed that all of the stated grounds of rejection have been properly traversed, accommodated, or rendered moot. Applicant therefore respectfully requests that the Examiner reconsider and withdraw all presently outstanding rejections. It is believed that a full and complete response has been made to the outstanding Office Action and the present application is in condition for allowance. Thus, prompt and favorable consideration of this amendment is respectfully requested.

If the Examiner believes that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at (248) 641-1600.

Respectfully submitted,

Dated: November 15, 2007

By: 
Timothy D. MacIntyre
Reg. No. 42,824

HARNESS, DICKEY & PIERCE, P.L.C.
P.O. Box 828
Bloomfield Hills, Michigan 48303
(248) 641-1600

TDM/med